

## GitHub Pages 简介

GitHub Pages 是一种静态站点托管服务，该服务直接从 GitHub 上的仓库获取 HTML，CSS 和 JavaScript 文件，可以选择在构建过程中运行这些文件，只需编辑、推送，网站的更改会在发布时生效。您可以将站点托管在 GitHub 的 `github.io` 域或您自己的自定义域上。

---

## GitHub Pages 网站的类型

GitHub Pages 站点有三种类型：项目站点、用户站点和组织站点。

用户站点“<https://<username>.github.io/>”的三级域名与其 GitHub 的登录帐号相同。用户站点的仓库“<username>/<username>.github.io”只能有一个。

项目站点“<https://<username>.github.io/<name>>”是用户站点的子目录。项目站点的数目没有限制。

---

## 在 GitHub 上构建自己的网站

### 1. 发布用户站点 (User sites)

步骤 1：创建仓库 (repository)

- 1) 前往 [GitHub](#) 创建一个名为“<username>.github.io”的仓库，其中<username>是 GitHub 的登录帐号。
- 2) 在 GitHub 上，导航到站点的仓库 (repo)：“<https://github.com/<username>/<username>.github.io>”，在仓库名称下，点击“Settings”。

或者前往“<https://github.com/<username>/<username>.github.io/settings>”。

- 3) 在“GitHub Pages”下，点击“Choose a theme”，选择主题后点击“Select theme”。

现在您的用户网站“<https://<username>.github.io>”已经自动发布，等待 5 分钟即可正常访问。

步骤 2：使用 GitHub Desktop 部署用户网站

GitHub Desktop 是 Git 的图形化工具。

- 1) 前往“<https://desktop.github.com/>”下载并安装 GitHub Desktop。
- 2) 在本地磁盘上创建一个用于与 GitHub 仓库 (repo) 同步的文件夹“E:\GitHub”。
- 3) 打开 GitHub Desktop，登录帐号。点击上方菜单栏“File” — “Clone repository”，从仓库 (repo) “<username>/<username>.github.io/”克隆文件到本地的“E:\GitHub\”目录下。
- 4) 删除“E:\GitHub\<username>.github.io\”目录下所有的文件。只保留名为“.git”的隐藏文件。
- 5) 前往“<http://jekyllthemes.org/>”，下载主题模板并解压至“E:\GitHub\<username>.github.io\”目录下。
- 6) 回到 GitHub Desktop，在“summary”一栏里标注，然后点击“Commit to main”，再点击“Push origin”，从本地磁盘推送至“<username>.github.io”仓库。

现在您对网站“<https://<username>.github.io>”的更改已经生效，等待 5 分钟刷新页面即可。

### 2. 发布项目站点 (Project sites)

步骤 1：创建仓库 (repository)

- 1) 前往 [GitHub](#) 创建一个名为“blog”的仓库。
- 2) 页面自动跳转至“<https://github.com/<username>/blog>”，新建分支 (Branches) “gh-pages”。

- 3) 点击“settings”，在“GitHub Pages”中找到生成的项目站点：“[https:// <username>.github.io/blog/](https://<username>.github.io/blog/)”。点击“Choose a theme”，选择主题并点击“Select theme”。
- 4) 返回“blog”仓库的设置页面：“<https://github.com/Dangiaon/blog/settings>”。点击“Branches”。
- 5) 展开“Default branch”（默认分支）下拉框，选择“gh-pages”然后点击“Update”。
- 6) 在弹出的对话框中点击“I understand, update the default branch.”。

现在您的项目网站“[https:// <username>.github.io/blog/](https://<username>.github.io/blog/)”已经发布，等待 5 分钟即可正常访问。

步骤 2：使用 GitHub Desktop 部署项目网站

GitHub Desktop 是 Git 的图形化工具。

- 1) 打开 GitHub Desktop，点击上方菜单栏“File” — “Clone repository”，从仓库“<username>/blog”克隆文件到本地的“E:\GitHub\”。
- 2) 删除“E:\GitHub\<username>.github.io\”目录下所有的文件。只保留名为“.git”的隐藏文件。
- 3) 前往“<http://jekyllthemes.org/>”，下载主题模板并解压至“E:\GitHub\blog”目录下。
- 4) 回到 GitHub Desktop，在“summary”一栏里标注，然后点击“Commit to main”，再点击“Push origin”，从本地磁盘推送至“blog”仓库。

现在您对网站“<https://<username>.github.io/blog/>”的更改已经生效，等待 5 分钟刷新页面即可。

---

## Fork 与 Clone；Pull 与 Fetch 的区别

Fork：从外部的仓库复制到自己的 GitHub 中。

Clone：从自己或外部的仓库克隆到本地的磁盘上。

- 1) Fork。从外部的仓库复制到自己的 GitHub 中。自己 GitHub 中的仓库是 Fork 源的一个分支。
- 2) Clone。从自己的仓库克隆到本地的磁盘上进行编辑。
- 3) Push origin。从本地磁盘推送到自己的 GitHub 仓库中。

获取更新然后合并。

```
1. git fetch origin master:tmp
2. git diff tmp
3. git merge tmp
```

上述过程等效于：

```
1. git pull origin master
```

---

## 目录结构（Jekyll）

- index.html 网站的主页。
- \_config.yml 配置文件。一般存放博主的信息。
- \_layouts 页面模板。一般有 default.html（默认页面）和 posts.html（博文页面）。
- \_includes 复用代码块。
- \_posts 博文。必须使用 markdown 编写。